

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-240129

(43) 公開日 平成10年(1998) 9月11日

(51) Int.Cl.⁶

G 0 9 C 5/00

H 0 4 N 1/387

識別記号

F I

G 0 9 C 5/00

H 0 4 N 1/387

審査請求 有 請求項の数 2 O L (全 10 頁)

(21) 出願番号 特願平9-247998

(22) 出願日 平成9年(1997) 9月12日

(31) 優先権主張番号 特願平8-348426

(32) 優先日 平8(1996)12月26日

(33) 優先権主張国 日本 (J P)

(71) 出願人 592073101

日本アイ・ビー・エム株式会社
東京都港区六本木3丁目2番12号

(72) 発明者 小出 昭夫

神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社東京基礎研究所内

(72) 発明者 森本 典繁

神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社東京基礎研究所内

(72) 発明者 清水 周一

神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社東京基礎研究所内

(74) 代理人 弁理士 坂口 博 (外1名)

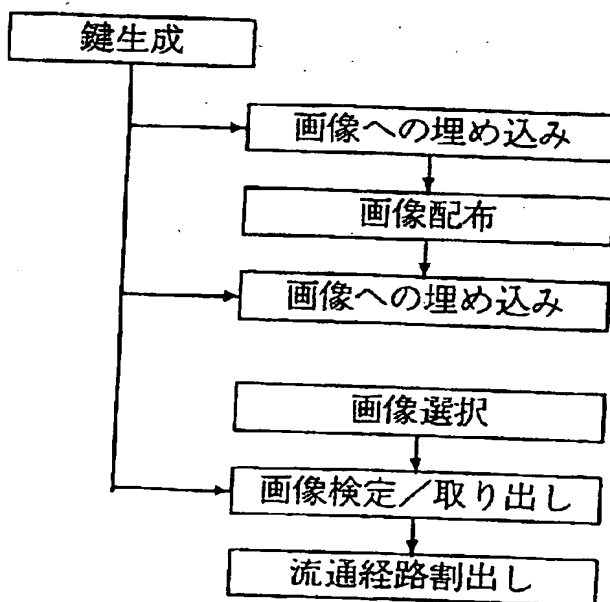
最終頁に続く

(54) 【発明の名称】 統計検定を用いたデータ・ハイディング方法及びデータ抽出方法

(57) 【要約】

【課題】 真の所有者がだれであるかを適切に証明でき、画素値等の特性値に対する操作量を適応的に決定することにより、埋め込み処理が施されたメディア情報の画質等の劣化を抑制すること。

【解決手段】 メディア・データ中にメッセージ・データを埋め込むデータ・ハイディング方法において、所定の値を有する鍵を特定の関数の入力とすることにより、二次鍵を求めるステップと、求められた二次鍵に基づいて、メッセージ・データを埋め込む位置を特定すると共に、この二次鍵に基づいて、複数の埋め込み用関数の中から、当該特定された位置において使用される埋め込み用関数を特定するステップと、特定された位置において、特定された埋め込み用関数に基づいて、メッセージデータを埋め込む処理を行うステップとを有するデータ・ハイディング方法である。



【特許請求の範囲】

【請求項1】メディア・データ中にメッセージ・データを埋め込むデータ・ハイディング方法において、所定の値を有する鍵を特定の関数の入力とすることにより、二次鍵を求めるステップと、

求められた前記二次鍵に基づいて、前記メッセージ・データを埋め込む位置を特定すると共に、前記二次鍵に基づいて、複数の埋め込み用関数の中から、当該特定された位置において使用される前記埋め込み用関数を特定するステップと、

前記特定された位置において、前記特定された埋め込み用関数に基づいて、メッセージデータを埋め込む処理を行うステップとを有することを特徴とするデータ・ハイディング方法。

【請求項2】メディア・データ中に埋め込まれたメッセージ・データを取り出すデータ抽出方法において、所定の値を有する鍵を特定の関数の入力とすることにより、二次鍵を求めるステップと、

求められた前記二次鍵に基づいて、前記メッセージ・データを埋め込む位置を特定すると共に、前記二次鍵に基づいて、複数の埋め込み用関数の中から、当該特定された位置において使用される前記埋め込み用関数を特定するステップと、

前記特定された位置内の情報を、前記特定された埋め込み用関数に基づいて指定された検出基本関数に与えるステップと、

前記検出基本関数の出力を検定関数に与えるステップと、

前記検定関数の出力に基づき、前記メッセージ・データが隠されているか否かを判定するステップとを有することを特徴とするデータ抽出方法。

【発明の詳細な説明】

【0001】

【発明の属する利用分野】本発明は、デジタル画像、デジタルビデオ、デジタルオーディオ等のメディア情報中に、その所有者や著作権に関する情報（メッセージ情報）を、非可視または非可聴の形で埋め込むデータ・ハイディング方法及びデータ抽出方法に関する。特に、統計的な検定を使ってメディア情報に対する埋め込み操作を適切に制御しながら、情報を非可視または非可聴の形で埋め込むデータ・ハイディング方法に関する。また、統計的な検定を使って、メディア情報が埋め込み処理が施されているか否かを判定し、この判定結果に基づいて、埋め込まれた情報を適切に取り出すデータ抽出方法に関する。

【0002】

【従来の技術】統計的手法を用いたデータハイディング方法として、以下のような技術が知られている。まず、画像データ中から二つの画素点列（それぞれの点列を以下、 $\{a_n\}$ と $\{b_n\}$ と呼ぶ）を選び出す。それぞれの画素点

列は、多数の画素点で構成されており、その数を n 個とする。そして、一方の点列 $\{a_n\}$ の n 個の画素値 $v(a_n)$ に一定値 c を加えると共に、他方の点列 $\{b_n\}$ の n 個の画素値 $v(b_n)$ から一定値 c を引くことにより、埋め込み操作を行う。

【数1】 $v_0(a_n) = v(a_n) + c$

$v_0(b_n) = v(b_n) - c$

【0003】画像データ中に埋め込み操作が施されているか否かの判定については、下式のように、 n 個の两点列の画素値の差に関して、その平均を計算し、その結果に基づいて判定する。

【数2】

$$\frac{1}{N} \sum_{n=1}^N (v'(a_n) - v'(b_n))$$

すなわち、統計的性質が出現する程度に多数の画素値に基づき、その差の平均値をとれば、何も加算処理が施されていない場合には、その値が0に収束することが期待される。一方、上記の加算処理が施されていれば、その平均値は一定値 $2c$ になることが期待される。従って、ある閾値を設けておいて、平均値が0と $2c$ のどちらの値に近いかに基づいて、埋め込み操作が行われているか否かを判断する。

【0004】埋め込まれる情報は1ビットで、二つの画素点列の位置は、埋め込み操作を行った本人のみ知る秘密とする。画素点列の位置を特定できない限り、埋め込み操作を施した本人以外の者は平均値 $2c$ という値を抽出することはできないので、この値を抽出できることが、そのデータの所有者証明となる。

【0005】

【発明が解決しようとする課題】しかしながら、この統計的手法を用いた従来技術の第一の問題点は、新たな埋め込み操作を施すことにより、データの所有者が誰であるかを容易に不明にできることである。すなわち、第三者が、所有者のみ知る画素点列中の画素値に操作を加えなくても、適当な別の二つの画素点列を選び出し、それらの画素値に埋め込み操作を施すことにより、別の点列において、新たに平均値 $2c$ を作り出すことができる。従って、このような処理を施した第三者が、自身が特定した点列から計算した平均値を理由に、自身が所有者であると偽って主張した場合、従来技術では誰が真の所有者であるかを証明することはできない。

【0006】また、第二の問題点は、埋め込み処理とそれによる画質の劣化に関する考察が十分になされていないことである。同じ値 c を加算・減算するにしても、ある画質の特性により、視覚的な画質の劣化に大きな差が生じることがある。従って、画質の特性によって、埋め込み操作量を適応的に変えることが好ましい。すなわち、一定値 c や画素点列中の画素数 N をを適応的に選択することが好ましい。

【0007】そこで、本発明の目的は、真の所有者がだれであるかを適切に証明できるデータ・ハイディング方法を提供することである。

【0008】また、本発明の別の目的は、画素値等の特性値に対する操作量を適応的に決定することにより、埋め込み処理が施されたメディア情報の画質等の劣化を抑制することである。

【0009】

【課題を解決するための手段】上記課題を解決するために、本発明の特徴の一つは、画素点列の選択するための鍵を使用する点にある。これにより、偽りの所有者証明を防ぐことができる。また、別の特徴として、データの埋め込みに際して、選択の画素点列のサイズ(N)及び特性値に対して操作すべき統計量の大きさ(c)を適応的に決定することである。そして、データの抽出に際しては、ある領域中に情報が埋め込まれているか、すなわち、当該領域中の特性値に演算処理が施されているか否かの判定のための信頼度を算出する。これらによって、画像に複数ビットの情報(著作者や配布IDなど)を非可視の形で埋め込み、また、画像から埋め込んだの情報を取り出し、画像の所有権の主張や配布経路を割り出すことができる。

【0010】具体的には、第1の発明は、メディア・データ中にメッセージ・データを埋め込むデータ・ハイディング方法において、所定の値を有する鍵を特定の関数の入力とすることにより、二次鍵を求めるステップと、求められた二次鍵に基づいて、メッセージ・データを埋め込む位置を特定すると共に、二次鍵に基づいて、複数の埋め込み用関数の中から、当該特定された位置において使用される前記埋め込み用関数を特定するステップと、特定された位置において、特定された埋め込み用関数に基づいて、メッセージデータを埋め込む処理を行うステップとを有するデータ・ハイディング方法を提供する。

【0011】また、第2の発明は、メディア・データ中に埋め込まれたメッセージ・データを取り出すデータ抽出方法において、所定の値を有する鍵を特定の関数の入力とすることにより、二次鍵を求めるステップと、求められた二次鍵に基づいて、メッセージ・データを埋め込む位置を特定すると共に、二次鍵に基づいて、複数の埋め込み用関数の中から、当該特定された位置において使用される埋め込み用関数を特定するステップと、特定された位置内の情報を、特定された埋め込み用関数に基づいて指定された検出基本関数に与えるステップと、検出基本関数の出力を検定関数に与えるステップと、検定関数の出力に基づき、メッセージ・データが隠されているか否かを判定するステップとを有するデータ抽出方法を提供する。

【0012】

【発明の実施の形態】図1は、画像の配布と鍵生成との

関係を示した図である。まず最初に鍵を生成し、鍵に基づき複数ビットの情報を画像に埋め込み配布する。もし、暗号化されて配布するなら、配布先で再度複数ビットの情報を解凍された画像に追加埋め込む。次に流通している画像を鍵に基づき検定し、著作権を侵害していないかを監視し侵害があった場合に取り出されたビット情報から流通経路を割出す。以下ではデジタルコンテンツへの埋込みがあるか否かの検定とビット情報の取り出しシステムの概要について先に説明し、次に複数ビット情報の検定/埋め込みシステムの概要について説明する。

また、本実施例における重要な概念である、検出基本関数と重ね書き、検定関数算出機能、偽の埋め込み証明、及び一方向性関数を利用した埋め込みについて詳細に説明する。

【0013】(1) 検定/取り出しシステム

検定/取り出しシステムは次から構成される。

- ・与えられた鍵から点列と検出基本関数を指定する機構
- ・点列と検出基本関数から各点で数値を求めその和を算出する機構
- ・検定関数を求める機構
- ・算出された和に検定関数を適用し、ビット情報とその確率的信頼度を求める機構

【0014】点列はデジタルコンテンツの埋め込み場所を指定するもので、デジタルコンテンツが一次元なら点列は一次元座標の列、デジタルコンテンツが二次元なら点列は二次元座標の列とする。点列を複数の集団に分け、各点列集団に1ビット情報を埋め込み、埋め込んだデジタルコンテンツから集団の数だけのビット長の情報を取り出す。点列の各点を x_{na} と記す。二重添字は点列が複数の集団に分かれていることを示すために便宜上付けたもので、aは所属する集団を表す。

【0015】検出基本関数は指定された点の近傍のデジタルデータを使って数値を算出する機構で、検出基本関数の具体的例は後述する。点列のすべての点で共通の検出基本関数を使っても良いが、鍵に基づいて複数の検出基本関数から各点で選択し異なっても良い。悪意をもった第三者によって埋め込み情報が消去されることを防ぐために、鍵に基づいて複数の検出基本関数から各点で選択し異なるようにする。以下、点 x_{na} での検出基本関数を f_{na} と記す。

【0016】点列と検出基本関数に基づいて各点で数値を求め、各集団(添字a)でのその和 s_a を算出するものとする。

【数3】

$$s_a = \sum_{n=1}^{N_a} f_{na}(x_{na})$$

ここで、 N_a は集団aに属する点の数で、集団ごとに異なっても良い。点列と検出基本関数の値は保持する必要がなく、鍵から次々と生成し、和の蓄積メモリーに加

算されしだい破棄すればよい。

【0017】検定関数とは、与えられたデジタルコンテンツに対し、点列や検出基本関数がランダムに選択されたとして、その和がある値より大きい確率、及び、小さい確率を与えるものとする。点列がN個のときの和がsより大きい確率を $E_+(N, s)$ と記し、sより小さい確率を $E_-(N, s)$ と記す。このとき、算出された和 s_a から、ビット情報とその確率的信頼度を次のように求める。

【数4】 $E_+(Na, s_a) > E_-(Na, s_a)$

上記関係を満足するならば、ビット0が埋め込まれているとし、 $E_+(Na, s_a)$ をその信頼度とする。逆に、

【数5】 $E_+(Na, s_a) < E_-(Na, s_a)$

ならば、ビット1が埋め込まれているとし、 $E_-(Na, s_a)$ をその信頼度とする。等号が成立するなら、ビット0と1の確率は同じ、すなわち、ビット情報の検出ができないものとする。もし、検定関数が次を満たすように構成されているならば、上記のビット情報とその確率的信頼度を求める機構はもっと簡単化される。

【数6】 $E_+(N, s) + E_-(N, s) = 1$

このとき、

【数7】 $E_+(Na, s_a) > 0.5$

ならば、ビット0が埋め込まれているとし、 $E_+(Na, s_a)$ をその信頼度とする。逆に、

【数8】 $E_+(Na, s_a) < 0.5$

ならば、ビット1が埋め込まれているとし、 $1 - E_+(Na, s_a)$ をその信頼度とする。また、上記のビット検出において、ビット0とビット1との検出規約を反転してもよい。

【0018】デジタルコンテンツへのビット情報が埋め込みがなされているか否かの判定は、このサブシステムの実際の利用者がその目的に応じて設定した信頼度を越えるかによって行う。上記の検定/取り出しシステムではデジタルコンテンツデータへのランダムアクセスが必要となるが、デジタルコンテンツデータをメモリー領域に保持せずにストリームとして処理するには、システムを次の構成にする。

- ・デジタルコンテンツのストリームデータに対して、その位置(点)と与えられた鍵からどの検出基本関数fを使い、どの和 s_a に加算するかを決定する機構

- ・検出基本関数fの値を一時的に求め、和 s_a に加算し蓄積する機構

- ・ストリームデータから検定関数を求めるに必要なデータを蓄積する機構

- ・ストリームデータすべて処理した後、検定関数を求め、蓄積された和 s_a に検定関数を適用し、ビット情報とその確率的信頼度を求める機構

【0019】(2)埋め込みシステム概要

埋め込みシステムは次から構成される。

- ・与えられた鍵から点列と検出基本関数を指定する機構
- ・検定関数を求める機構

- ・確率的信頼度と検定関数からビット情報に埋めるに必要な和の大きさを求める機構

- ・検出基本関数の和が所定値を越えるように、非可視性(非可聴性)を保ちながら点列の各点の近傍のデジタルデータに操作を加える機構

【0020】点列と検出基本関数をそれぞれ x_{na} 、 f_{na} と記す。埋め込みシステムで鍵から生成される点列の集団の数については、以下に説明するように、操作によってデジタルコンテンツの重要な統計的特性を変えないようにするため、検定/取り出しシステムで用いる数より一般には多くとること、また、個々の集団aに属する点の数 N_a は、ビット情報を信頼度を高く埋め込みたい集団については大きくとる。検定関数は、 $E_+(N, s)$ 、 $E_-(N, s)$ と記す。集団aに信頼度 p_a 以上でビット情報を埋め込むとき、ビット0に対しては検出基本関数の和 s_a が

【数9】 $E_+(Na, s_a) > p_a$

になるよう埋め込み、ビット1に対しては検出基本関数の和 s_a が

【数10】 $E_-(Na, s_a) > p_a$

になるよう埋め込む。ここで、信頼度 p_a は0.5より大きいとする。すなわち、ビット0に対しては検出基本関数の和が $E_+(Na, s_a) = p_a$ なる s_a より小さく、ビット1に対しては検出基本関数の和が $E_-(Na, s_a) = p_a$ なる s_a より大きくとる。なお、上記のビット埋め込みにおいて、ビット0とビット1との埋め込み規約を反転してもよい。

【0021】検定関数はデジタルコンテンツの統計的特性に依存するので、埋め込みシステムでの点列の集団の数は検定/取り出しシステムでの数より多くとり、統計的特性の変更を打ち消すよう余分の集団に操作を加える。特に、デジタルコンテンツ全体の検出基本関数の平均値が、操作前と変わらないように余分の集団の検出基本関数の和の目標値を定め、埋め込み操作を行う。打ち消しのための余分の集団は1個でも複数でもかまわない。一つのビットを埋め込むたびに、それを打ち消す埋め込みを行い、ビット埋め込み集団と余分の集団を同数にとる。

【0022】埋め込み操作は、非可視性(非可聴性)を保ちながら、点列の各点 x_{na} の近傍の点の値を操作して行う。埋め込み前のデジタルコンテンツの検出基本関数の和が s_a^0 とすると、 $\Delta s_a = s_a - s_a^0$ が各集団の目標変更幅である。もし、点列の各点での変更幅を同一に取れば、 $\Delta s_a / N_a$ が各点での検出基本関数値の目標変更幅である。

【0023】非可視性(非可聴性)を保ちながら改ざん耐性の持つよう埋め込むために、点列の各点 x_{na} での変更幅を同一でなく、埋め込みが目立ち易い領域では小さい変更幅で、目立たない領域では大きい変更幅で各点の近傍の点の値を操作して行う。デジタルコンテンツの埋め込み点列の各点での非可視性(非可聴性)の指標を算出し、その指標に応じて検出基本関数値の目標変更幅を決

め、その点の近傍の点の値を操作する。

【0024】非可視性(非可聴性)の指標とは、与えられた点 x での近傍の点でのデジタルコンテンツの値より算出される量で、次のタイプがある。

- ・比例型指標: 各点 x での変更幅が指標 $g(x)$ に比例していれば同程度の可視性(可聴性)を与える指標。
- ・閾値型指標: 点 x での変更幅が指標 $g(x)$ より小さければ非可視性(非可聴性)である指標。
- ・混合型指標: 上記の組み合わせ。

比例型指標では、比例係数 r を次のようにして求め、 $rg(x_{na})$ を各点 x_{na} での目標変更幅とする。

【数11】

$$r = \Delta s_a / \sum_{n=1}^{N_a} g(x_{na})$$

閾値型指標では、

【数12】

$$|\Delta s_a| \leq \sum_{n=1}^{N_a} g(x_{na})$$

ならば、 $|\Delta s_a|$ を越えるまで順に各点 x_{na} で変更幅 $g(x_{na})$ の操作を Δs_a の符号の向きに加え、越えた時点で変更操作を止めるか、または、すべての点 x_{na} で

【数13】

$$g(x_{na})\Delta s_a / \sum_{n=1}^{N_a} g(x_{na})$$

の変更を行う。また、

【数14】

$$|\Delta s_a| \geq \sum_{n=1}^{N_a} g(x_{na})$$

ならば、各点 x_{na} で変更幅

【数15】

$$g(x_{na}) + (|\Delta s_a| - \sum_{n=1}^{N_a} g(x_{na})) / N_a$$

の操作を Δs_a の符号に応じて行う。混合型指標については、上記の組み合わせで埋め込みを行う。

【0025】(3) 検出基本関数と重ね書き
鍵は点列と検出基本関数を指定する。ここでは検出基本関数とそれに対応する埋め込み操作、重ね書き手法について説明する。検出基本関数は指定された点の近傍の点でのデジタルデータを使って数値を計算する機構である。検出基本関数を f_α で記し、 α を複数の検出基本関数を区別するための添字とする。

【0026】まず、係数の和が零の線形フィルタについて説明する。検出基本関数の形は原理的には自由だが、デジタルコンテンツは通常整数値の配列で与えられるので、整数値を入力として整数値を出力とすることと、また、非可視性を効率良く満たすために、検出基本関数の値の分布がその平均値のまわりに集中していて検出基本関数の値を変更するに必要な近傍の点で値の変更幅が小さいものが望ましい。後者の条件を正確に述べると、デジタルコンテンツ全体での検出基本関数の標準偏差を σ とすると、検出基本関数の値を σ だけ増減するに必要な近傍の点での値の変更幅の平均が、各点での値のデジタルコンテンツ全体での標準偏差より小さい検出基本関数が望ましい。

【0027】このような性質を持つ検出基本関数として下式のような線形フィルタがある。

【数16】

$$f_\alpha(x) = \sum_y F_\alpha(y) v(x+y)$$

ここで、 $v(x+y)$ は点 x から y だけ移動した点でのデジタルコンテンツの値で、フィルタの係数 $F_\alpha(y)$ は整数で

【数17】

$$0 = \sum_y F_\alpha(y)$$

とする。デジタル画像やデジタルビデオでは x 、 y を二次元ベクターとする。係数の和を零としているのは、埋め込み情報がその点のデジタルデータの絶対値に依存するのではなく、その点の周りのデジタルデータの振舞い、すなわち、相対値に依存するようにするためである。例えば、デジタル画像のもっとも簡単な線形フィルタとして次を取り上げてみる。検出基本関数 f_{s0} の係数を

【数18】($F_{s0}(0,0)$, $F_{s0}(1,0)$) = (1, -1)

で与え、検出基本関数 f_{s1} の係数を

【数19】($F_{s1}(0,0)$, $F_{s1}(0,1)$) = (1, -1)

で与える。図2はこれらの検出基本関数の特性をテスト画像に試した表である。それぞれの検出基本関数の標準偏差は、画素値の標準偏差よりかなり小さい。従って、画素値そのものを検出基本関数とするより、 f_{s0} 、 f_{s1} を検出基本関数として使用した方が小さい変更幅で済むと期待できる。

【0028】JPEG、MPEGや対応するためには、DCT変換する 8×8 のブロックと調和する幅、高さの線形フィルタ、すなわち、 4×4 、 4×8 、 8×4 、 8×8 、 16×8 、 8×16 、または 16×16 の線形フィルタを検出基本関数に用いる。例えば、次の 8×8 フィルタ f_{j0} 、 f_{j1} 、 f_{j2} 、 f_{j3} を使用する。

【数20】

$$F_{J0}(j, k) = \begin{pmatrix} 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & -1 & 0 & 0 & 1 & 1 \\ -1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 \\ -1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 1 & 0 & 0 & -1 & -1 \end{pmatrix}$$

【数21】

$$F_{J1}(j, k) = \begin{pmatrix} 0 & 0 & -1 & -1 & 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & -1 & 0 & 0 & 1 & 1 \\ -1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 \\ -1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 1 & 0 & 0 & -1 & -1 \\ 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \end{pmatrix}$$

【数22】

$$F_{J2}(j, k) = \begin{pmatrix} 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 1 & 0 & 0 & -1 & -1 \\ -1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 \\ -1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & -1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

【数23】

$$F_{J3}(j, k) = \begin{pmatrix} 0 & 0 & -1 & -1 & 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & -1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 1 & 0 & 0 & -1 & -1 \\ -1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 \\ -1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

【0029】点列の各点で検出基本関数値の目標変更値 $\Delta f_{\alpha}(x) = f_{\alpha}(x) - f_{\alpha}(x)$ を非可視性（非可聴性）の指標を用いて決める。ここでは、その目標値を実現するために埋め込み点列の点の近傍の各点でのデジタルデータへの操作を線形フィルタの場合について具体的に記述する。以下で、埋め込み点列の点 x に対し、その近傍の各点 $x+y$ での変更値を $w(x, y)$ と記す。すなわち、点 $x+y$ におけるデジタルデータの値 $v(x+y)$ を $v(x+y) + w(x, y)$ に変更するとする。

【0030】検出基本関数 f_{α} の線形フィルタの係数 $F_{\alpha}(y)$ に対し、

40 【数24】

$$\sum_y F_{\alpha}(y) G_{\alpha}(y) = D_{\alpha} > 0$$

なる係数 $G_{\alpha}(y)$ を定め、

【数25】

$$d(x) = \Delta f_{\alpha}(x) / D_{\alpha}$$

より、 $d(x)$ を求め、変更値を

【数26】

$$w(x, y) = d(x) G_{\alpha}(y)$$

で求める。最も簡単な係数 $G_{\alpha}(y)$ の選択は $G_{\alpha}(y) = F_{\alpha}(y)$ である。

50 $d(x)$ の絶対値が1より大きい場合、 $d(x) G_{\alpha}(y)$

(y) の値を少し変化させて緩やかに値が変化するように均し、非可視性を強める。例えば、 $G_{J1}(j, k) = F_{J1}(j, k)$

$$d(x)G_{J1}(j, k) = \begin{pmatrix} 0 & 0 & -4 & -4 & 0 & 0 & 4 & 4 \\ 0 & 0 & -4 & -4 & 0 & 0 & 4 & 4 \\ -4 & -4 & 0 & 0 & 4 & 4 & 0 & 0 \\ -4 & -4 & 0 & 0 & 4 & 4 & 0 & 0 \\ 0 & 0 & 4 & 4 & 0 & 0 & -4 & -4 \\ 0 & 0 & 4 & 4 & 0 & 0 & -4 & -4 \\ 4 & 4 & 0 & 0 & -4 & -4 & 0 & 0 \\ 4 & 4 & 0 & 0 & -4 & -4 & 0 & 0 \end{pmatrix}$$

の代わりに、その値を均した

【数28】

$$w(x, y) = \begin{pmatrix} 0 & 0 & -3 & -4 & 0 & 0 & 4 & 3 \\ 0 & 0 & -4 & -4 & 0 & 1 & 3 & 4 \\ -3 & -4 & 0 & 0 & 4 & 3 & 1 & 0 \\ -4 & -4 & 0 & 0 & 4 & 4 & 0 & 0 \\ 0 & 0 & 4 & 4 & 0 & 0 & -4 & -4 \\ 0 & 1 & 3 & 4 & 0 & 0 & -4 & -3 \\ 4 & 3 & 1 & 0 & -4 & -4 & 0 & 0 \\ 3 & 4 & 0 & 0 & -4 & -3 & 0 & 0 \end{pmatrix}$$

を使用する。

【0031】複数のメッセージを重ね書きで埋め込むために次の手法を用いる。

- ・直交型埋め込み
- ・階層型埋め込み

ここで、直交型埋め込みとは、互いに独立性の高い検出基本関数を用いてビットを重ね書きするもので、線形フィルタの場合、

【数29】

$$\sum_y F_\alpha(y) G_\beta(y) = D_\alpha \neq 0 \quad \text{if } \alpha = \beta \\ = 0 \quad \text{otherwise}$$

という直交する係数の組みを用いて行うものである。 α と β が異なれば、 $w(x, y) = d(x) G_\beta(y)$ で行った埋め込みが、係数 $F_\alpha(y)$ で与えられる検出基本関数 f_α で検出されることはない。

【0032】また、階層型埋め込みとは、適用される近傍の領域のサイズが異なる検出基本関数の間で、小さいサイズの検出基本関数の適用域を大きいサイズの検出基本関数の適用域に形式的に拡大したとき、両者の間の独立性が高いものをいう。例えば、 2×1 サイズの線形フィル

$$P_N(s) = \sum_{f_1} \sum_{f_2} \dots \sum_{f_{N-1}} p(f_1) p(f_2) \dots p(f_{N-1}) p(s - f_1 - f_2 \dots - f_{N-1})$$

これをもちいて検定関数は次のように求めることができる。

【数34】

に対し、 $d(x) = 4$ のとき、

【数27】

ター f_{S0} は、次の 2×2 サイズの線形フィルタに拡張できる。

【数30】

$$F_{S0.0}(i, j) = \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix}$$

$$F_{S0.1}(i, j) = \begin{pmatrix} 0 & 0 \\ 1 & -1 \end{pmatrix}$$

これらは線形フィルタ

【数31】

$$F_{SS}(i, j) = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}$$

と前述の直交型埋め込みの意味で直交する。従って、 f_{SS} と f_{S0} とは重ね書きができる。

【0033】ここでは、検定関数算出について説明する。デジタルコンテンツが指定されたとき、検出基本関数 f_α が値 f をとる度数をコンテンツ全体で数え、頻度分布(ヒストグラム) $h(f)$ を作成する。これから、検出基本関数 f_α が値 f である確率 $p(f)$ を次で算出する。

【数32】

$$p(f) = h(f) / \sum_f h(f)$$

コンテンツのすべての点について検出基本関数 f_α の値を計算するのではなく、ランダムに選択された点に関して検出基本関数 f_α の値を計算し頻度分布を作成しても、選択された点の選択が十分多ければ、実用上かまわない。得られた確率 $p(f)$ から、 N 個の検出基本関数の和が s である確率 $P_N(s)$ を下式により求める。

【数33】

$$E_+(N, s) = \sum_{s' > s}^{13} P_N(s')$$

$$E_-(N, s) = \sum_{s' < s} P_N(s')$$

【0034】検定関数近似的算出

以下では、上記の検定関数を効率良く求める方法として、検定関数を検出基本関数の統計モーメント、すなわち、巾乗の平均値 $\langle f^n \rangle$ から近似的に求める方法について説明する。ここで、統計モーメントは

【数35】

$$\langle f^n \rangle = \sum_x \sum_{\alpha} f_{\alpha}(x)^n / \sum_x \sum_{\alpha} 1$$

で算出するものとする。以下に述べるように、ヒストグラム $h(f)$ から確率 $P_N(s)$ を計算する必要がないので、

$$E_+(N, s)^{(0)} = \frac{1}{\sqrt{2\pi N \langle f^2 \rangle_c}} \int_s^{\infty} ds' \exp\left(-\frac{(s' - N \langle f \rangle)^2}{2N \langle f^2 \rangle_c}\right)$$

$$E_-(N, s)^{(0)} = \frac{1}{\sqrt{2\pi N \langle f^2 \rangle_c}} \int_{-\infty}^s ds' \exp\left(-\frac{(s' - N \langle f \rangle)^2}{2N \langle f^2 \rangle_c}\right)$$

【0035】近似検定関数の補正項 $E_+(N, s)^{(n)}$ 、 $E_-(N, s)^{(n)}$ を次で算出する。

【数39】

$$E_+(N, s)^{(n)} = -Q_n(N, s) P_N(s)^{(0)}$$

$$E_-(N, s)^{(n)} = Q_n(N, s) P_N(s)^{(0)}$$

簡潔さのために

【数40】

$$u = s - N \langle f \rangle$$

及び

$$Q_2(N, s) = \frac{u}{N} \left[\frac{(\langle f^4 \rangle_c - 3 \langle f^2 \rangle_c^2)}{4! \langle f^2 \rangle_c^2} (w - 3) + \frac{\langle f^3 \rangle_c^2}{2! 3! \langle f^2 \rangle_c^3} (w^2 - 10w + 15) \right]$$

【0036】検定関数を補正項を加えて $E_+^{(0)} + E_+^{(1)} + E_+^{(0)} + E_+^{(1)} + E_+^{(2)}$ 、 $E_-^{(0)} + E_-^{(1)} + E_-^{(0)} + E_-^{(1)} + E_-^{(2)}$ で評価したとき、負になったとき、零と置き換える。

【0037】検定関数厳密算出

近似無しに確率 $P_N(s)$ を効率的に求めるには漸化式

【数44】

$$P_{N+N'}(s) = \sum_{s'} P_N(s') P_{N'}(s - s')$$

を使う。例えば、 $N=2M$ には、上記の漸化式を M 回繰り返せば良い。欠点はメモリーサイズである。裁判などで近似が忌み嫌われるときに使用する。

【0038】偽りの埋め込み証明

デジタルコンテンツが指定されたとき、検出基本関数 f_{α} が値 f をとる度数をコンテンツ全体で数え、頻度分布(ヒストグラム) $h(f)$ を作成し、頻度分布 $h(f)$ に操作を

「モリ」も演算量も少なく済む。式を簡潔にするため、 $\langle f^n \rangle_c = \langle (f - \langle f \rangle)^n \rangle$ と置く。すなわち、

【数36】

$$\langle f \rangle_c = 0$$

$$\langle f^2 \rangle_c = \langle f^2 \rangle - \langle f \rangle^2$$

$$\langle f^3 \rangle_c = \langle f^3 \rangle - 3 \langle f^2 \rangle \langle f \rangle + 2 \langle f \rangle^3$$

$$\langle f^4 \rangle_c = \langle f^4 \rangle - 4 \langle f^3 \rangle \langle f \rangle + 6 \langle f^2 \rangle \langle f \rangle^2 - 3 \langle f \rangle^4$$

とする。このとき、十分に大きい N に対して、

10 【数37】

$$P_N(s)^{(0)} = \frac{1}{\sqrt{2\pi N \langle f^2 \rangle_c}} \exp\left(-\frac{(s - N \langle f \rangle)^2}{2N \langle f^2 \rangle_c}\right)$$

と近似できる。これより、検定関数は次で与えられる。

【数38】

$$w = \frac{(s - N \langle f \rangle)^2}{N \langle f^2 \rangle_c}$$

と置くと、 $Q_n(N, s)$ は次で与えられる。

【数42】

$$Q_1(N, s) = \frac{\langle f^3 \rangle_c}{3! \langle f^2 \rangle_c} (w - 1)$$

【数43】

加えて点列に対応する頻度分布 $h_a(f)$ を作成し、以下のような点列を求める。

・点列上での検出基本関数の和が目標値を越えるようにする

・点列上での検出基本関数の和が目標値に近付くようにする

・二つの点列上での検出基本関数の差の和が目標値に近付くようにする

【0039】一方向関数による鍵のシステムがないと、このシステムによって、ビット情報が埋め込まれていないデジタルコンテンツに対して、埋め込みがなされたという偽りの証拠を作成することができる。以下にこの発明を具体的に述べる。

【0040】デジタルコンテンツ全体での検出基本関数値の度数分布を $h(f)$ とし、点列集団 a での検出基本関数値の度数分布を $h_a(f)$ とすると、すべての f に対し、

【数45】

$$0 \leq h_a(f) \leq h(f)$$

を満たし、点列集団aでの検出基本関数値の和は

【数46】

$$s_a = \sum_f f h_a(f)$$

で与えられ、点列集団aの点の個数は

【数47】

$$N_a = \sum_f h_a(f)$$

で与えられる。

【0041】このとき、従来技術で説明したビット抽出条件式(数1)は、さらに、

【数48】

$$s_a/N_a = -s_b/N_b = c$$

がほぼ成立し、

【数49】

$$N_a = N_b = N \text{ and } h_a(f) + h_b(f) \leq h(f)$$

である $h_a(f)$ 、 $h_b(f)$ を求めることとなる。また、本発明の検出関数によるビット抽出条件は、検出関数が N_a が大きければ近似的に正規分布に近づくことに注目すれば、ビット1のときには、下式が成立する。

【数50】

$$s_a/\sqrt{N_a} \geq c$$

一方、ビット0のときには、下式が成立する。

【数51】

$$s_a/\sqrt{N_a} \leq -c$$

が成立し、

【数52】

$$\sum_a h_a(f) \leq h(f)$$

である $h_a(f)$ を求めることとなる。

【0042】偽りの埋め込み証明を作成するシステムは、最初のスキャンでデジタルコンテンツから度数分布 h を作成し、次に $h(f)$ より上記を満たす度数分布 h_a を作成し、次のスキャンでデジタルコンテンツから $f = f(x)$ なる点を $h_a(f)$ 個だけ選び出し、その集まりを偽の埋め込みの点列とする。

【0043】全体の度数分布 h から集団の度数分布 h_a の作成は、 $h_a(f) = 0$ を初期値とし、和 $\sum_f f h_a(f)$ が値 N_a となるまで、以下に述べる所定の規則のもとに f を算出し、 $h(f)$ が正なら1だけ減らし、 $h_a(f)$ を1だけ増す基本操作を、すべての a について繰り返す。 a についてのループが内側である。

【0044】検出関数によるビット抽出条件の度数分布 h_a を作成するには、ビット1の埋め込みに使われる度数分布については大きい f から順に上記の基本操作を行ない、ビット0の埋め込みに使われる度数分布については

小さい f から順に上記の基本操作を行なう。

【0045】従来技術で説明したビット抽出条件式(数1)は、次のように行なう。 $\Delta a_0(+)=0$ 、 $\Delta a_0(-)=0$ を初期値とし、以下、

【数53】

$$f_{na} = c + \Delta_n(+)$$

に近い f_{na} を求め h_a に基本操作を行ない、

【数54】

$$\Delta_{n+1}(+) = c + \Delta_n(+)-f_{na}$$

10 とし、

【数55】

$$f_{nb} = -c + \Delta_n(-)$$

に近い f_{nb} を求め h_b に基本操作を行ない、

【数56】

$$\Delta_{n+1}(-) = -c + \Delta_n(-)-f_{nb}$$

とする。最終誤差は、 $\Delta N(+)/N$ と $\Delta N(-)/N$ で与えられる。

【0046】一方向性関数を利用した埋め込み

上述のように、埋め込みのないデジタルコンテンツから点列の選択で所定のビット情報をあたかも埋め込まれていたかのように取り出すことができる。従って、デジタルコンテンツから所定のビット情報を取り出す点列を知っているだけでは、そのデジタルコンテンツの所有者であるか否かを証明できない。この問題の解決方法として、特定のデジタルコンテンツに特定の点列で埋め込んでいることを公正な第三者機関に事前に登録して置くことも考えられるが、この方法では次のような欠点を持つ。

・各埋め込みごとに登録しなければならず、登録業務に経費がかかる。

30

・埋め込み点列を第三者機関に登録するので、それだけ、埋め込み点列の機密が暴露される危険性が増し、埋め込み情報が消去される危険にさらされる。

40

【0047】偽りの所有者証明の問題を解決するものとして、整数値(以下「鍵」と呼ぶ)から一方向性関数を使って埋め込み箇所の指定する手法を公開し、「鍵」を本人のみが知る秘密とし、ビット情報を埋め込む方式及びシステムを発明として請求する。この方式を採用する限り、特定のデジタルコンテンツから偽りの所有者証明の点列を求めることができても、一方向性関数の特性のためその点列を導出する「鍵」を算出できないので、秘密鍵を知ることによる偽の所有者証明を行なうことができない。

50

【0048】鍵から一方向性関数を使って埋め込み箇所の指定する点列を生成する方法は、鍵から一方向性関数を使って「二次鍵」を生成し、「二次鍵」から点列を生成することにより実現できる。ここで鍵も「二次鍵」も非負の整数値だから通常の一方向性関数を使用してもかまわない。鍵から一方向性関数を使って「二次鍵」を生成することにより、鍵から一方向性関数を使って埋め込

み箇所の指定する方式を実現する。

【0049】次に「二次鍵」から点列を生成する方法について具体的に説明する。デジタルコンテンツをN個の領域に分割し番号をふる。いま、その番号をnで記す。各領域nを、検出基本関数適用が適用されるM個の部分領域に分割し番号をふる。また、検出基本関数はL個の中から選択されるものとする。このとき、二次鍵は、各領域nからその部分領域の番号 m_n 、及び、その部分領域に適用する検出基本関数 f_{ln} を選択する。すなわち、二次鍵(非負整数) kが部分領域と検出基本関数を指定する整数

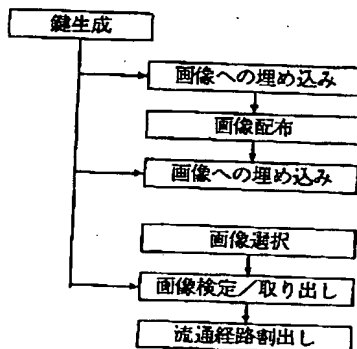
【数57】

$$j = \sum_{n=0}^{N-1} L^n M^n (m_n + M l_n)$$

を生成するものとする。二次鍵の取りうる範囲(kのビット)長は整数jの取りうる範囲(jのビット長)より通常短い。従って、 k_0 を二次鍵とすると、 k_i から k_{i+1} を順に算出し、次に

【数58】

【図1】



$$j = \sum_{i=0}^{18} K^i k_i$$

を算出し、部分領域と検出基本関数を指定する。ここで、Kは非負整数 k_i の上限である。 k_i から k_{i+1} の算出機構は、一方向性関数でも、通常の演算でも良い。一対一関数であるものが望ましい。部分領域の数M、検出基本関数の数Lを2の巾乗にとることにより、整数jらの m_n 、 l_n の算出がビット演算でできる。

【0050】

【効果】このように本発明によれば、真の所有者がだれであるかを適切に証明でき、画素値等の特性値に対する操作量を適応的に決定することにより、埋め込み処理が施されたメディア情報の画質等の劣化を抑制することができる。

【図面の簡単な説明】

【図1】実施例における画像の配布と鍵生成との関係を示した図である。

【図2】検出基本関数の特性をテスト画像に試した結果を示す表である。

【図2】

サンプル画像	画素値の標準偏差	f_{30} の標準偏差	f_{31} の標準偏差
200000 RED	68.3396	9.4510	9.0329
200000 GREEN	59.6419	9.5007	9.0248
200000 BLUE	60.9283	9.7633	9.2070
200001 RED	58.7429	16.2642	17.1555
200001 GREEN	54.6955	16.3756	17.1682
200001 BLUE	53.0666	15.9694	16.8779
200002 RED	52.5882	4.4696	8.5511
200002 GREEN	45.1880	4.1811	8.1316
200002 BLUE	37.9568	4.0256	7.8376
200011 RED	16.0885	3.2931	3.4074
200011 GREEN	18.8486	3.3313	3.4857
200011 BLUE	21.4928	3.4299	3.6076

フロントページの続き

(72) 発明者 小林 誠士

神奈川県大和市下鶴間1623番地14 日本ア
イ・ビー・エム株式会社東京基礎研究所内